

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1. (Currently Amended) A processor module for a player station operable by a player to play a game thereon, comprising:

~~a processor capable of executing a software program to generate a simulation of the game;~~

a storage memory accessible by the processor;

an interface facility communicable with the processor and with at least one peripheral device;

a unique identification code associated with the processor module; and

a security module co-operable with the processor, the security module being arranged to decrypt an encrypted software program to recover an identification code therefrom, to enable execution of the software program by the processor to generate a simulation of the game when the ~~program is encrypted with~~ recovered identification code matches the unique identification code associated with the processor module, and to disable execution of the software program when the ~~software program is encrypted with a code that is different from~~ recovered identification code does not match the unique identification code associated with the processor module.

2. (Original) A processor module as claimed in claim 1 in which the security module also disables execution of the software program when the software program is unencrypted.

3. (Original) A processor module as claimed in claim 1 in which the unique identification code is stored in a protected area of the storage memory.

4. (Original) A processor module as claimed in claim 3 in which the protected area of the storage memory is a read-only memory.

5. (Original) A processor module as claimed in claim 1 in which the interface facility is an input/output circuit connected to the processor by means of an input/output bus.

6. (Original) A processor module as claimed in claim 1 in which the at least one peripheral device is any one of a display monitor, a magnetic card reader, a banknote validator, an array of pushbuttons, a coin acceptor, a ticket reader, a numeric keypad, a printer and a counter.

7. (Original) A processor module as claimed in claim 1 in which communication between the processor and the at least one peripheral device is encrypted.

8. (Original) A processor module as claimed in claim 1 in which the processor module includes a random number generator.

9. (Original) A processor module as claimed in claim 8 in which the random number generator is a hardware random number generator.

10. (Original) A processor module as claimed in claim 9 in which the storage memory includes a still further portion that is removable.

11. (Original) A processor module as claimed in claim 10 in which the removable portion of the storage memory is a flash memory module.

12. (Currently Amended) A processor module as claimed in ~~claim 1~~ claim 1 in which the processor includes a network interface that provides access to a communication network.

13. (Original) A processor module as claimed in claim 12 in which the communication network is the Internet.

14. (Original) A processor module as claimed in claim 1 in which the processor module also includes a number of interface ports.

15. (Original) A processor module as claimed in claim 14 in which the number of interface ports include any one or more of a serial communication port and a port conforming to the Universal Serial Bus standard.

16. (Original) A method for configuring a processor module for a player station operable by a player to play a game thereon, comprising the steps of:

providing a software program executable to generate a simulation of the game;

obtaining a unique identification code associated with the processor module;

encrypting the software program remotely from the processor module as a function of the unique identification code;

transferring the encrypted software program to the processor module;

decrypting the encrypted software program to obtain a decrypted identification code therefrom; and

enabling execution of the encrypted software program by the processor module when the decrypted identification code is the same as the unique identification code of the processor module and disabling execution of the encrypted software program by the processor module when the decrypted identification code is different from the unique identification code of the processor module.

17. (Original) A method as claimed in claim 16 that includes a step of also disabling execution of the software program by the processor module when the software program is unencrypted.

18. (Original) A method as claimed in claim 16 in which the unique identification code is stored securely in a protected area of a storage memory associated with the processor module.

19. (Original) A method as claimed in claim 16 in which execution of the encrypted software program is enabled for a predetermined period of time.

20. (Original) A method as claimed in claim 19 in which execution of the encrypted software program is re-enabled upon the occurrence of a predetermined event.

21. (Original) A method as claimed in claim 16 that includes a step of jointly encrypting a plurality of different software programs remotely from the processor module as a function of the unique identification code, each one of the plurality of software programs being executable to produce a simulation of a different game.

22. (Original) A method as claimed in claim 21 in which the jointly encrypted plurality of software programs are transferred to the processor module.

23. (Original) A method as claimed in claim 22 in which the jointly encrypted plurality of software programs are decrypted to obtain a decrypted identification code therefrom, and execution of any selected one of the jointly encrypted plurality of software- programs is enabled when the decrypted identification code is the same as the unique identification code of the processor module, and execution of all of the jointly encrypted plurality of software programs is disabled when the decrypted identification code is different from the unique identification code of the processor module.

24. (Original) A system for customisation and distribution of software, comprising:
a number of player stations, each player station being operable by a respective player to play a corresponding game, each player station being associated with a unique identification code;
a repository containing a number of different software programs, each software program being executable by at least one of the number of player stations to generate a simulation of a different game;
a download server communicable with the repository;

a communication network enabling communication between the download server and each one of the number of player stations;

encryption means operable to encrypt, remotely from the number of player stations, a selectable one of the number of different software programs contained in the repository as a function of the unique identification code of a selectable one of the number of player stations, the download server being responsive to the encryption means to download the encrypted software program to the particular player station whose unique identification code was used for encryption; and

a security module associated with the particular player station, the security module being capable of decrypting the downloaded encrypted software program to obtain therefrom a decrypted identification code and enabling execution of the downloaded encrypted software program by the particular player station when the decrypted identification code is the same as the unique identification code of the particular player station, and disabling execution of the downloaded encrypted software program by the particular player station when the decrypted identification code is different from the unique identification code of the particular player station.

25. (Original) A system as claimed in claim 24 in which the security module also disables execution of the software program by the particular player station when the software program is unencrypted.

26. (Original) A system as claimed in claim 24 in which the security module enables execution of the downloaded encrypted software program for a predetermined period of time.

27. (Original) A system as claimed in claim 26 in which the security module re-enables execution of the downloaded encrypted software program upon the occurrence of a predetermined event.

28. (Original) A system as claimed in claim 24 in which the encryption means jointly encrypts, remotely from the number of player stations, a plurality of different software programs as a function of the unique identification code of the selectable one of the number of player stations, each one of the plurality of software programs being executable to produce a simulation of a different game.

29. (Original) A system as claimed in claim 28 in which the download server is responsive to the encryption means to download the jointly encrypted plurality of software programs to the particular player station whose unique identification code was used for encryption.

30. (Original) A system as claimed in claim 29 in which the security module decrypts the jointly encrypted plurality of software programs to obtain a decrypted identification code therefrom and enables execution of any selected one of the jointly encrypted plurality of software programs by the particular player station when the decrypted identification code is the same as the unique identification code of the particular player station and disables execution of all of the jointly encrypted plurality of software programs by the player station when the decrypted identification code is different from the unique identification code of the particular player station.

31. (Original) A system as claimed in claim 24 in which each player station has an associated storage memory.

32. (Original) A system as claimed in claim 31 in which the unique identification code of the player station is stored securely in a protected area of the storage memory.

33. (Original) A system as claimed in claim 32 in which the protected area of the storage memory is a read-only memory.

34. (Original) A system as claimed in claim 24 in which each player station includes a processor and a number of peripheral devices.

35. (Original) A system as claimed in claim 34 in which the number of peripheral devices include any one or more of a display monitor, a magnetic card reader, a banknote validator, an array of pushbuttons, a coin acceptor, a ticket reader, a numeric keypad, a printer and a counter.

36. (Original) A system as claimed in claim 34 in which communication between the processor and the at least one peripheral device is encrypted.

37. (Original) A system as claimed in claim 24 in which the processor module includes a random number generator.

38. (Original) A system as claimed in claim 37 in which the random number generator is a hardware random number generator.

39. (Original) A system as claimed in claim 24 in which the storage memory includes a portion that is removable.

40. (Original) A system as claimed in claim 39 removable portion is a flash memory module.

41. (Original) A system as claimed in claim 24 in which the communication network is the Internet.

42. (Original) A system as claimed in claim 34 in which the processor module also includes a number of interface ports.

43. (Original) A system as claimed in claim 42 in which the number of interface ports includes any one or more of a serial communication port and a port conforming to the Universal Serial Bus standard.

44. (Original) A method for the customisation and distribution of software, comprising the steps of:

providing a number of player stations, each player station being operable by a respective player to play a corresponding game, each player station being associated with a unique identification code;

providing a repository containing a number of different software programs, each software program being executable by at least one of the number of player stations to generate a simulation of a different game;

encrypting, remotely from the number of player stations, a selected one of the number of different software programs contained in the repository as a function of the unique identification code of a selected one of the number of player stations;

downloading the encrypted software program to the particular player station whose unique identification code was used for encryption;

decrypting the downloaded encrypted software program to obtain therefrom a decrypted identification code; and

enabling execution of the downloaded encrypted software program by the particular player station when the decrypted identification code is the same as the unique identification code of the particular player station, and disabling execution of the downloaded encrypted software program by the particular player station when the decrypted identification code is different from the unique identification code of the particular player station.

45. (Original) A method as claimed in claim 44 that includes a step of also disabling execution of the software program by the particular player station when the software program is unencrypted.

46. (Original) A method as claimed in claim 44 in which execution of the downloaded encrypted software program is enabled for a predetermined period of time.

47. (Original) A method as claimed in claim 46 in which execution of the downloaded encrypted software program is re-enabled upon the occurrence of a predetermined event.

48. (Original) A method as claimed in claim 44 in which a plurality of different software programs are jointly encrypted, remotely from the number of player stations, as a function of the unique identification code of the selected one of the number of player stations, each one of the plurality of software programs being executable to produce a simulation of a different game.

49. (Original) A method as claimed in claim 48 in which the jointly encrypted plurality of software programs are downloaded to the particular player station whose unique identification code was used for encryption.

50. (Original) A method as claimed in claim 49 in which the jointly encrypted plurality of software programs are decrypted to obtain a decrypted identification code therefrom, and execution of any selected one of the jointly encrypted plurality of software programs by the particular player station is enabled when the decrypted identification code is the same as the unique identification code of the particular player station, and execution of all of the jointly encrypted plurality of software programs by the player station is disabled when the decrypted identification code is different from the unique identification code of the particular player station.

51. (Original) A method as claimed in claim 44 in which the unique identification code of the player station is stored securely in a protected area of a storage memory associated with the player station.

52. (Original) A system for the distribution of software, comprising:

a repository containing a number of different executable software programs;

a download server communicable with the repository;

a number of processor modules, each processor module being identified by means of a unique identification code and being operable to execute any one of the number of different software programs contained in the repository;

receiving means for receiving a request for a licence to execute a desired combination of at least one software program contained in the repository on at least one of the number of processor modules, the licence request containing at least one selectable identification code-to-software program mapping;

payment means for receiving a fee for the requested licence;

encryption means responsive to payment of the fee to encrypt the particular software program contained in the at least one selectable mapping as a function of the identification code in the mapping;

a download facility operable to download the encrypted particular software program to the particular processor module whose identification code was used for encryption; and

a security module associated with the particular processor module, the security module being capable of decrypting the downloaded encrypted software program to obtain therefrom a decrypted identification code and enabling execution of the downloaded encrypted software

program by the particular processor module when the decrypted identification code is the same as the unique identification code of the particular processor module, and disabling execution of the downloaded encrypted software program by the particular processor module when the decrypted identification code is different from the unique identification code of the particular processor module.

53. (Original) A system as claimed in claim 52 in which the at least one selectable mapping is a one-to-one mapping.

54. (Original) A system as claimed in claim 53 in which the licence request contains a plurality of different one-to-one mappings, each unique processor module identification code being contained in only one such mapping.

55. (Original) A system as claimed in claim 54 in which the encryption means is responsive to payment of the fee to encrypt the particular software program contained in each one of the plurality of different one-to-one mappings as a function of the identification code in that mapping.

56. (Original) A system as claimed in claim 55 in which the download facility downloads each encrypted software program to the particular processor module whose identification code was used for encryption.

57. (Original) A system as claimed in claim 52 in which the at least one selectable mapping is a many-to-one mapping.

58. (Original) A system as claimed in claim 57 in which the licence request contains a number of different many-to-one mappings, each unique processor module identification code being contained in only one such mapping.

59. (Original) A system as claimed in claim 58 in which the encryption means is responsive to payment of the fee to encrypt the particular software program contained in each one of the different many-to-one mappings with each one of the plurality of identification codes in that mapping to obtain separate encrypted instances of the same software program.

60. (Original) A system as claimed in claim 59 in which the download facility downloads each encrypted instance of a software program to the particular processor module whose identification code was used for encryption in that instance.

61. (Original) A system as claimed in claim 52 in which the security module also disables execution of the software program when the software program is unencrypted.

62. (Original) A system as claimed in claim 52 in which each processor module has an associated storage memory.

63. (Original) A system as claimed in claim 52 in which the unique identification code of the processor module is stored securely in a protected area of the storage memory.

64. (Original) A system as claimed in claim 63 in which the protected area of the storage memory is a read-only memory.

65. (Original) A system as claimed in claim 52 in which the processor module is interfaceable to at least one peripheral device.

66. (Original) A system as claimed in claim 65 in which the at least one peripheral device is any one or more of a display monitor, a magnetic card reader, a banknote validator, an array of pushbuttons, a coin acceptor, a ticket reader, a numeric keypad, a printer and a counter.

67. (Original) A system as claimed in claim 66 in which communication between the processor module and the at least one peripheral device is encrypted.

68. (Original) A system as claimed in claim 52 in which the processor module includes a random number generator.

69. (Original) A system as claimed in claim 68 in which the random number generator is a hardware random number generator.

70. (Original) A system as claimed in claim 64 in which the storage memory includes a portion that is removable.

71. (Original) A system as claimed in claim 70 in which the removable portion of the storage memory is a flash memory module.

72. (Original) A system as claimed in claim 52 in which the processor module includes a network interface for providing access to a communication network.

73. (Original) A system as claimed in claim 72 in which the communication network is the Internet.

74. (Original) A system as claimed in claim 52 in which the processor module also includes a number of interface ports.

75. (Original) A system as claimed in claim 74 in which the number of interface ports includes any one or more of a serial communication port and a port conforming to the Universal Serial Bus standard.

76. (Original) A method for the distribution of software, comprising the steps of:
providing a repository containing a number of different executable software programs,
providing a number of processor modules, each processor module being operable to execute any

one of the number of different software programs contained in the repository, and identifying each processor module by means of a unique identification code;

receiving a request for a licence to execute a desired combination of at least one software program contained in the repository on at least one of the number of processor modules, the licence request containing at least one selectable identification code-to-software program mapping; receiving a fee for the requested licence;

encrypting, in response to payment of the fee and remotely from the number of processor modules, the particular software program contained in the at least one selectable mapping as a function of the identification code in the mapping;

downloading the encrypted particular software program to the particular processor module whose identification code was used for encryption;

decrypting the downloaded encrypted software program to obtain therefrom a decrypted identification code; and

enabling execution of the downloaded encrypted software program by the particular processor module when the decrypted identification code is the same as the unique identification code of the particular processor module, and disabling execution of the downloaded encrypted software program by the particular processor module when the decrypted identification code is different from the unique identification code of the particular processor module.

77. (Original) A method as claimed in claim 76 in which the at least one selectable mapping is a one-to-one mapping.

78. (Original) A method as claimed in claim 77 in which the licence request includes a

plurality of different one-to-one mappings, each unique processor module identification code being contained in only one such mapping.

79. (Original) A method as claimed in claim 77 that includes a step of encrypting, in response to payment of the fee, the particular software program contained in each one of different one-to-one mappings as a function of the identification code in that mapping.

80. (Original) A method as claimed in claim 79 in which each encrypted software program is downloaded to the particular processor module whose identification code was, used for encryption.

81. (Original) A method as claimed in claim 76 in which at least one selectable mapping is a many-to-one mapping.

82. (Original) A method as claimed in claim 81 in which the licence request includes a number of different many-to-one mappings, each unique processor module identification code being contained in only one such mapping.

83. (Original) A method as claimed in claim 82 which includes a step of encrypting, in response to payment of the fee, the particular software program contained in each one of the different many-to-one mappings with each one of the plurality of identification codes in that mapping to obtain separate encrypted instances of the same software program.

84. (Original) A method as claimed in claim 83 in which each encrypted instance of a software program is downloaded to the particular processor module whose identification code was used for encryption in that instance.

85. (Original) A method as claimed in claim 76 in which execution of the software program is disabled when the software program is unencrypted.

86. (Original) A method as claimed in claim 76 in which the unique identification code of the processor module is stored securely in a protected area of a storage memory associated with the processor module.

87. (Original) A method as claimed in claim 76 that includes a step of interfacing the processor module to at least one peripheral device.

88. (Original) A method as claimed in claim 87 in which communication between the processor module and the at least one peripheral device is encrypted.